Deployment API of the server to carry out the deployment. In another example, for the delivery of database content, the corresponding engine connects with a specific target database instance, and creates and executes SQL statements to update the data in this instance. Other examples are factories for file system deployment and for deployment of database structure definitions into a database instance. These examples show how diverse the deployment can be in individual cases and that SDM **14** is not restricted to only certain forms of deployment.

[**0031**] SDM **14** implements a general framework for the deployment of software, a framework that can include all deployment procedures for multiple server types. One or more Java interfaces are provided by SDM **14**, which describe the API for the deployment of particular software types on corresponding server types. The server type factories and classes implement these interfaces. Example server types implemented for SDM **14** include deployment of J2EE applications on a J2EE engine; transfer of standalone Java developments (jars) to a JRE (Java Runtime Environment); and delivery of database contents. Having a general framework, one has the option of creating more implementations of these interfaces for additional server types.

[**0032**] SDM **14** also provides a programming API that can be used by using SDM **14** to define target systems, prepare deployment for SDAs, submit specific parameter values, and carry out deployments.

[**0033**] When deploying SDAs, SDM **14** stores data in a repository **16**. The repository **16** includes a list of software delivery catalogs, SDAs, and products installed on the target system **10**. The repository **16** also includes information about SDA dependencies, platform information, and fingerprints of every file used in applications running on the target system **10**. SDM **14**, in conjunction with the repository **16**, recognizes dependencies between archives and provides support when shared applications are installed and maintained. SDM **14** supports resolution of intersystem dependencies, automatically extracting dependency information from the archives to be installed and comparing them against actual system states.

[**0034**] SDM **14** provides a browser-like GUI for local software administration that enables browsing, installation and patch application, for deployment of software updates to any target system.

[**0035**] The repository **16** contains information about all known logical target systems. All software deliveries are typed and, through a type mapping, target systems within a local deployment environment are found. SDAs are prepared for deployment into all logical targets that fulfill the dependencies the SDA requests. The deployer can deselect targets manually.

[**0036**] As shown in **FIG. 3**, an illustrative deployment process **200** includes presenting (**202**) an SDM user interface, which can be a graphical user interface (GUI), to the user deployer. The process **200** receives (**204**) user input for adding or deleting SWCs or SDAs to or from a deployment list.

[**0037**] The process **200**, using SDM **14**, generates (**206**) one or more deployment proposals from a manifest and SDA deployment descriptor data and displays (**208**) this information in an information window. The process **200** receives (**210**) user input specifying a deployment configuration. The deployment configuration is optionally displayed (**212**) in a navigation tree window. The process **200** notifies (**214**) the user that SDM **14** is ready to perform deployment. Upon the user's direction, SDM performs (**216**) the deployment. The process **200** displays (**218**) a success message on the user interface upon successful completion of deployment and updates (**222**) the repository **16** upon completion of the deployment.

[**0038**] In the case of updates of already deployed archives, SDM allows the reuse of the deployment parameters of a previous archive version. For example, in the J2EE case, if there are no changes in the deployment parameters between two versions of an SDA, SDM deploys the newer version without any human interaction by extracting the parameters from the J2EE target server and using them for the second deployment.

[**0039**] More specific information about the SDA and its contents will now be provided. SDA archives can be packed using a variety of tools like WinZip or jar.

[**0040**] The following attributes describing the component of the software contained in an SDA are contained in the standard manifest of the SDA:

[**0041**] Specification—Title

[**0042**] Specification—Version

[**0043**] Specification—Vendor

[**0044**] Implementation—Title

[**0045**] Implementation—Version

[**0046**] Implementation—Vendor

[**0047**] Implementation—Vendor-Id

[**0048**] As illustrated in **FIG. 4**, the following attributes are contained in the supplementary manifest **54** of an SDA:

[**0049**] 1. Attributes **41** describing compatibility of the SDA with different SDM versions:

[**0050**] Manifest—Version

[**0051**] SDM—Compatibility—Version

[**0052**] 2. Software type **42** of the software contained in the SDA:

[**0053**] softwaretype

[**0054**] 3. Name **43** of the SDA deployment descriptor contained in the SDA:

[**0055**] deployfile

[**0056**] 4. List of dependencies **44** on other SDAs:

[**0057**] dependencies

[**0058**] 5. Attributes **45** for each file contained in the SDA and in recursively contained SDAs:

[**0059**] Name

[**0060**] _md5fingerprint